NAME

x10config - Configuration file format for HEYU, an X-10 CM11A serial interface control program

DESCRIPTION

Heyu is a program for controlling an X-10 "CM11A" home control device. See the *heyu*(1) man page for usage information.

The configuration file contains several critical pieces of information that the *heyu* program needs in order to function, plus a number of user options. These are all specified by keywords known as **directives** which are explained below. In general, directives may appear in any order in the configuration file, however see the exception to this rule under the heading for the ALIAS directive.

The configuration file, named x10config, is normally stored in the (hidden) subdirectory .heyu under your home directory, i.e., $HOME/.heyu/x10config

For system-wide use, the configuration file may alternatively be stored as file x10.conf in the directory /etc/heyu (or as otherwise defined in Configure for your operating system).

The configuration file may be located in an additional subdirectory level under either of the directories above. The additional subdirectory level can have a name of your choice, for example "mysub". In this case you must specify the name of the subdirectory with the HEYUSUB environment variable in order for Heyu to find it, for example HEYUSUB=mysub. Somewhat more convenient however is to locate the configuration file in a subdirectory named simply /0 ... /9, e.g. $HOME/.heyu/3/x10config, in which case Heyu can be instucted to use the appropriate directory with the command line option ´-0´ ... ´-9´.

Finally, the configuration file may be stored in any arbitrary location on your hard drive with any arbitrary filename, but the full pathname will have to be specified either with the ´-c´ Heyu command line option or with the X10CONFIG environment variable.

If the HEYUSUB environment variable is NOT set or the ´-0´ ... ´-9´ not specified, then the search will start with $HOME/.heyu/x10config and continue with /etc/heyu/x10.conf, stopping with the first one it finds. If not found, Heyu will exit with an error message.

If the HEYUSUB environment variable IS set or the ´-0´ ... ´-9´ is specified, the search for the configuration file will start with $HOME/.heyu/<subdirectory>/x10config, then, if not found, will continue looking for /etc/heyu/<subdirectory>/x10.conf. Examples:

```
$HOME/.heyu/mysub/x10config
/etc/heyu/mysub/x10.conf
/etc/heyu/4/x10.conf
```

If not found under either of these places, Heyu will exit with an error message.

(The main reason for implementing the subdirectory feature is to enable Heyu to be used for more than one CM11A interface, with separate configuration and other files for each. If you have only one CM11A interface, you probably won´t find a need to bother with it.)

The directory in which the configuration file is located MUST be writable by Heyu.

Case sensistivity:
The following entities ARE case-sensitive: Filenames; Pathspecs; Serial ports (and the keyword "dummy" used when there is no CM11A connected);
Heyu and shell commands; Alias, Scene, Usersyn, Script, and Launcher labels.

The following entities are NOT case sensitive: Names of directives and values of directives other than those listed above; Housecode letters.

TTY directive

The TTY directive is the most important. Syntax is simply the word TTY, a space and then the full path name for the serial port to which the CM11A is attached. /dev/ttyS0 would be the first serial port (corresponding to COM1 under MS-DOS). /dev/cua0 will also work.
Examples:

```
TTY /dev/ttyS0
TTY /dev/cua2
TTY /dev/usb/ttyUSB0  (implies a USB-Serial adapter)
```

To configure Heyu for a CM10A interface (instead of a CM11A or CM12), append the keyword "CM10A", e.g.:

```
TTY /dev/ttyS0  CM10A
```

Note: If you´ve started Heyu before configuring for the CM10A, you must stop and start it up again. I.e., ´heyu stop´, then ´heyu start´.

TTY dummy directive

Setting the directive

```
TTY dummy
```

tells Heyu there is no CM11 powerline interface connected and Heyu commands which attempt to communicate with the interface will result in a quick error message. However commands which involve only the heyu_engine and heyu_aux daemons will continue to function, e.g., setting and reading flags or countdown timers, and executing scripts launched by RF signals forwarded from heyu_aux. (Like the name of the serial port it replaces, the word "dummy" is case sensitive.)

HOUSECODE directive

The housecode directive indicates the housecode for which the CM11A is to store in its internal registers the on/off/dim status of individual units when signals are sent or received over the AC power line. The ´heyu reset´ command (with no housecode parameter supplied) will program the CM11A to use the housecode provided by this directive. (It will not do this automatically.)

Example:

```
HOUSECODE   D
```

The default for this directive is housecode A.

ALIAS directive

An alias associates a label (front_porch) with an X10 device (A10) so that we don´t have to remember that the front porch light is house code A, device 10. In other words, we can use a mnemonic to represent a device.

There are a few other directives in which the user may choose to reference an alias label in place of a housecode|unitcode address. In these cases the alias directive must appear before the other directive in the configuration file in order to be resolved. Therefore it´s a good idea to group all the alias directives at or near the beginning of the file.

Alias labels can refer to one or more X10 devices with the same housecode, however Heyu´s monitor will display the alias label only if the alias references a single device address.

The format for an alias is:

```
ALIAS  Label  Housecode|Devices  [Module_type [Module option]]
```

where the Housecode letter and Device string are concatenated.

Label can be any contiguous string (maximum length 32 characters) of alphanumeric characters or period ´.´, hyphen ´-´, or underscore ´_´, except that the the hyphen and underscore may not be the first character in the string. The label may not be the word ´macro´.

Alias labels are case sensitive, so that ´kitchen´ and ´Kitchen´ represent two distinct aliases.

Housecode is a case-insensitive letter A-P (or a-p).
Devices can be 4 things:

1. A single number (example 1)
2. A group of numbers separated by commas (example 1,2,14,15)
3. A range separated by a dash (example 1-8)
4. A combination of 2 and 3 (example 1-8,14,15)

Module_type is optional and defines for Heyu the attributes of the particular kind of module assigned to the Housecode|Unit.
Heyu uses this information to determine what state the module should be in when the module receives any particular command. E.g., it will know that an appliance module will never be in a dimmed state even though it may be sent a dim command.

Unlike many popular software programs which attempt to do your thinking for you, Heyu does not decide what X10 command to send based on the Module_type defined for the specific housecode|unit address or alias. It does only what you tell it to do, and defining a Module_type does not restrict Heyu from sending a command which the module may not support. E.g., if you tell Heyu to send a dim command to an appliance module, it will do so without question. But the physical module will not respond and the Heyu state will reflect that fact.

The following generic module types and/or specific model numbers are currently recognized. The names are not case-sensitive:

*** WARNING - WARNING - WARNING ***

The X-10 LM465 Lamp Module and WS467 Wall Switch Module were redesigned in 2007 but the model numbers, appearance, and packaging remained unchanged. The new designs, designated here as module types LM465-1 and WS467-1, have quite different features and characteristics from the original designs. Among other differences, they support a subset of Extended Code commands. They can be distinguished by executing ´heyu xon Hu´ and ´heyu xoff Hu´ at the command prompt. The redesigned modules will respond and turn respectively On and Off; the original modules won't respond.

Among the potential problems, there´s an apparent design error in the redesigned WS467 Wall Switch firmware making it not backwards compatible with the original design. A standard X10 bright or dim or dimb signal won´t turn it On from the Off state unless preceded by an on signal.

LM465-1 (LM-1) - Redesigned LM465 lamp module.
LM (StdLM LM465 PLM01 PLM03) - Original standard X-10 lamp module.
LM12 () - Marmitek standard X-10 lamp module.
LMS (?) - X-10 2-way lamp module (responds status on/off).
AM (StdAM AM486 AM466 PAM01 PAM02) - Standard X-10 appliance module.
AM12 () - Marmitek standard X-10 appliance module.
SR227 (PA011) - X-10 socket module.
AMS (RR501 PAT01) - X-10 2-way switch (responds status on/off).
Some 2-way modules can be configured to automatically return a StatusOn/Off whenever they are turned On/Off. For these, the module type parameter "DEFER" instructs Heyu not to update the stored states of the module with the On/Off signal but let the ensuing StatusOn/Off signal do it. (Otherwise the changed state following the StatusOn/Off would always be unchanged.)
RAIN8II () WGL model Rain8II 2-way irrigation controller which can be configured to automatically respond with a StatusOn/Off signal when turned On/Off.
WS467-1 (WS-1) - Redesigned WS467 wall switch module.
WS (StdWS WS467) - Standard X-10 wall switch module.
LW10U () - Marmitek dimming wall switch module.

WS12A (XPD3) - X-10 dimming wall switch module.
WS13A (XPS3) - X-10 non-dimming wall switch module.
LM15A (PSM04) - X-10 Socket Rocket screw-in non-dimming lamp module.
LM15 () - Marmitek screw-in non-dimming lamp module.
AM14A (AM15A PAM21 PAM22) - X-10 2-way appliance module (Extended codes).
LM14A (PLM21) - X-10 2-way lamp module (Extended codes).
SL1AM (?) - SwitchLinc 1-way switch (Preset codes).
SL2AM (?) - SwitchLinc 2-way switch (Preset codes).
SL1LM (?) - SwitchLinc 1-way dimmer (Preset codes).
SL2LM (SL2380W) - SwitchLinc 2-way dimmer (Preset codes).
LL1LM (?) - LampLinc 1-way dimmer (Preset codes)
LL2LM (LL2000STW) - LampLinc 2-way dimmer (Preset codes)
RS114 () ACT user-programmable 2-way switch module.
With this module, support for AllUnitsOff, AllLightsOn, and AllLightsOff signals is user programmable. Module type parameters AUF, ALO, and ALF match Heyu's support with the module's programming.
RF234 () ACT user-programmable 2-way switch module. This 230V module has similar to but not identical characteristics with the RS114. Based on data from a single device, it appears the RS234 does not maintain its address after being turned On or Off.
AMEXC16 (?) - Module of the appliance type with exclusive addressing: An on command turns On only the last unit addressed. All other modules of this type on the same housecode turn themselves Off and become unaddressed.
AMEXC8 (RAIN8) - Modules of the appliance type with exclusive addressing within groups of 8 units, i.e., 1-8, 9-16. Other modules on the same housecode and within the same group of units turn themselves Off when the last addressed unit in that group is turned On. A typical device of this type is the WGL Rain8 Irrigation Controller.
AMEXC4 (XM10A XM13A) - Similar to AMEXC8 but with exclusive addressing within groups of 4, i.e., 1-4, 5-8, 9-12, 13-16. Typical modules of this type are X-10´s XM10A and XM13A camera power supplies.
PR511 () - X-10 PR511 2-way Motion Sensing floodlight
NONE - No module; supports no X10 commands.
REMOTE2 - Not actually a module, it can be used as a ´target´ for X10 commands sent by a transmitter with only On/Off codes (like a keychain remote or motion sensor), when there is no physical module assigned to that Housecode|Unit.
REMOTE4 - Like the above, but for a four function transmitter like a PalmPad which can send Dim and Bright codes in addition to On and Off.
REMOTE6 - Like the above, but for a six function transmitter like the Mini- or Maxi-Controller which can send On, Off, Dim, Bright, LightsOn, and AllOff codes.

Note that the above two REMOTEs function a little differently than lamp modules. The state for an On (or a LightsOn for REMOTE6) code will always be recorded as fully On regardless of prior state, and the state for Dim/Bright codes can range from

fully Off through fully On and vice-versa. This allows their use in scripts for controlling things other than modules.

REMOTE3 - Similar to REMOTE4 except that the On command is ignored by the state engine except for addressing the particular unit in the housecode - it does not change the stored brightness level. The Off command does set the brightness level to zero and the Bright and Dim commands make incremental changes in this level within the range fully Off to fully On. (The Bright and Dim signals sent by RF remotes do not include a unit code. This module type allows using the On signal to address a particular unit without changing the stored brightness level and then using the Dim and Bright commands for some analog control purpose via a script.)

REMOTEP - Again not actually a module, but can be used as a target for transmitters which send only the Preset (1-32) codes.

SHUTTER (SW10) - Shutter and shade controller which support Extended Code Type 0 commands. The Marmitek SW10 Shutter Motor Controller is the only module known to support these commands and Marmitek keeps it a secret. This Heyu model supports Extended Type 0 functions 0x01, 0x02, 0x03, 0x04, and 0x0B. The Heyu model also supports the standard On and Off commands, but not Dim or Bright, to which the shutter module is reported to react in an unpredictable manner.

VDATA - A virtual module which can store a data byte as if it were a (raw) brightness level 0-255. Data is written to this virtual module only with the ´heyu vdata HU <data>´ command - it is unaffected by any other X10 command.

The following security transmitter models are included for use with RF auxiliary input from a W800RF32A or RFXCOM receiver.

DS10A - (PDS01) X-10 DS10A Security Door/Window Sensor
DS90 - (DS18-1) X-10 DS90 Sec Door/Window Sensor (2 chan)
DS18-1 - () ElekHomica DS18 Sec Door/Window Sensor (2 chan)
DS18 - () ElekHomica DS18 Sec Door/Window Sensor (1 chan)
MS10A - (PMS01) X-10 MS10A Security Motion Sensor (See section "MS10A WARNING" in man page x10aux(5))
SH624 - (PSR01) X-10 Security Remote
KR10A - (PKR02, KR21) X-10 Security Keyfob Remote.
KR18 - (KR18E) Marmitek Security Keyfob Remote.
SD10 - () Marmitek SD10 Smoke Detectors
SD90 - () Marmitek SD90 Smoke Detectors
BMB-SD18 - () BMB SD18 Smoke Detector
MS90 - (MS18E) Marmitek MS90 Security Motion Sensor
EH-CWSD10 - () ElekHomica EH-CWSD10 Smoke Detector

EH-WD210 - () ElecHomica EH-WD210 Water Detector
GB10 - () Marmitek GB10 Glass Break Detector
SVDATA - () Generic X10 Security Remote (all ´vdata´)

Most (but not all) security sensors transmit a 16-bit security ID which is detectable with the RFXCOM RF receiver operating in variable length packet mode (the default). For backward compatibility, the directive ´SECURID_16 NO´ instructs Heyu to ignore the upper 8-bits.

Some security sensors, notably the Aux channel of the DS90 Door/Window sensor, have a firmware bug whereby a parity bit used to check the upper 8 bits of the 16-bit security ID is incorrect in every other RF repetition, which means that half the signals are discarded as noise. The configuration directive ´SECURID_PARITY NO´ instructs Heyu to disregard the parity bit. It's use is less risky than throwing away half the signals, but if none of your sensors exhibit this bug there's no need to use it. Configuration directive ´DISPLAY_RAW_RF NOISE´ can be used to determine whether the bug exists.

Unlike the larger SH624 security remote, the smaller KR10A, KR18, and KR21 security keyfob remotes have no physical Away/Home or Max/Min switches, and by default transmit the swAway and swMin flags when the Arm button is pressed. Appending the parameters SWHOME and/or SWMAX to the alias directive for these keyfob remotes results in the flags being decoded respectively as swHome and/or swMax.

Alternatively, appending the parameter DUMMY to the above security keyfob remote alias directive results in the signals transmitted by the Arm and Disarm button presses being decoded instead as Alert and Clear, thus allowing the remote to be used for a user-defined purpose unrelated to Arm and Disarm.

Module type SEC_IGNORE can be used to ignore signals from X10 Security sensors which may not be under your control, e.g., signals from a nearby neighbor´s sensor. An unused Housecode/Unit address must be sacrificed. Specify the Security IDs for one or more sensors to be ignored.
Example:
ALIAS Neighbor_Sensors P6 SEC_IGNORE 3C 4E 2A

The following entertainment transmitter model is included for use with RF Auxiliary input from a W800RF32A, RFXCOM, or MR26A receiver.

UR81A - (UR61A) X-10 Entertainment Universal Remote.

The following module types are used to override the settings defined by the TRANSCEIVE and RFFORWARD directives for specific units and functions within a housecode. Each requires one of the parameters TRANSCEIVE, RFFORWARD, or RFIGNORE be specified following the module name in the ALIAS directive.

PALMPAD - Controls RF On, Off, Dim, Bright.
KEYCHAIN - Controls RF On and Off
ONLYON - Controls RF On
ONLYOFF - Controls RF Off
MS12, *MS13*, *MS14*, *MS16* - Controls RF On and Off

(The MSxx module types are similar to the KEYCHAIN module type but are defined as "sensors" and as such will be listed in the table displayed by ´heyu show sensors´.)

Example: If housecode C is set to be transceived, specifying:
ALIAS XMMS_Control C1-6 KEYCHAIN RFFORWARD
allows using the On/Off buttons C1-6 on a RF remote to launch scripts controlling the XMMS audio player software on the PC without the delay of a powerline signal. The other unit buttons continue to transceive powerline signals to lamps and appliances.

PLCSENSOR - A target for a transmitter which sends
X10 On and Off signals over the power line and which has
a "heartbeat", i.e., it periodically retransmits its current
state. The Heyu config directives HIDE_UNCHANGED and
INACTIVE_TIMEOUT apply to this module type as they do for

RF security sensors.

The following module type decodes RF signals transmitted from RFXSensor modules and received by a W800RF32A/AE or RFXCOM RF receiver, and maps them to a Housecode|Unit address.

RFXSENSOR - A decoder for RFXSensor transmitters and external sensors. Two parameters are required, the base_address of the sensor and a mnemonic for the type of sensor, e.g., Temperature/Humidity, Temp/Barometric Pressure, etc. The mnemonic will be one of: TH, TB, TV, TP, TT, or T.

Example:

ALIAS Basement L9 RFXSENSOR 0x20 TH

The RF signals sent by the sensor with base_address 0x20 will be decoded as RFX Temperature and Relative Humidity functions received (RCVA) at Housecode|Unit L9.
See man page x10rfxsensors(5) for full details.

The following module types decode RF signals transmitted by RFXMeter sensors. Each requires as a parameter the ID address of the sensor.

RFXPOWER - Electric Watt-Hour meter.
RFXWATER - Water meter.
RFXGAS - Gas meter.
RFXPULSE - Pulse meter.
RFXCOUNT - General counter.

Example:

ALIAS MyWaterMeter C9 RFXWATER 0x04

Module type DIGIMAX decodes RF signals transmitted by the DigiMax 210 wireless thermostat and received by the 433.92 MHz RFXCOM X10 RF receiver in variable length packet mode. Its parameter is the 16-bit ID of the thermostat.

Module types ORE_xxx decode signals transmitted by various Oregon remote sensors and received by the 433.92 MHz RFXCOM X10 RF receiver in variable length packet mode. See man page x10oregon(5) for full descriptions.

Module type ELS_ELEC1 decodes signals transmitted by the Electrisave CM113 Electricity Monitor sensor and received by the 433.92 MHz RFXCOM X10 RF receiver in variable length packet mode. The optional directive "ELS_VOLTAGE <voltage>" specifies a nominal AC voltage to be multiplied by the measured current to display an apparent power. See page X10oregon(5) for more information.

Module type OWL_ELEC2 decodes signals transmitted by the OWL CM119 Electricity Monitor sensor and received by the 433.92 MHz RFXCOM X10 RF receiver in variable length packet mode. The directive "OWL_VOLTAGE <voltage>" specifies the nominal AC voltage to be used to compute the Power and Energy values. See page X10oregon(5) for more information.

SwitchLinc and LampLinc modules can be manually configured to always turn on to some level less than the maximum (32), or always resume the previous setting. If you've configured your modules in either of these ways, you must inform Heyu of the fact by adding the module option ´ONLEVEL nn´ or ´ONLEVEL RESUME´.

NOTE: In order for the Heyu state engine to properly track the state of modules with the ´resume´ feature (LM14A and SwitchLinc/LampLinc option), its memory must be synchronized with the module´s memory. This can be done by sending the module an X10 command setting it at a known level greater than the ´Off´ level, or by having it respond to a status request with a preset/xstatus level greater than the ´Off´ level.

Examples:

ALIAS hall_light B7
ALIAS patio_lights B2,4-6
ALIAS porch_light D1 WS467

ALIAS livingroom_lamp H7 SL2LM ONLEVEL 20

It is advisable to define single-unit aliases for all housecode|units in addition to any multiple-unit aliases which may be convenient for use in Scenes/Usersyns/Macros, and to define a Module_type only for single-unit aliases.
(Once a Module_type is defined, it applies to that same housecode|unit however used, e.g., in the above example Heyu will understand that D1 is a WS467 dimmer when either ´heyu turn porch_light on´ or ´heyu turn D1 on´ is executed, or if D1 is also included in a multiple-unit alias like D1,2-5,7)

Modules not defined for a Housecode|unit in an ALIAS directive are assigned to be the default module, which is a standard X-10 lamp module. This can be changed with the DEFAULT_MODULE directive.

Note: Versions of Heyu prior to 2 used a different format for aliases: No ALIAS directive was used and the Housecode letter and Units were separated by whitespace, e.g., simply ´hall_light B 7´. For compatibility, Heyu version 2 will still accept this format, however its use is discouraged and deprecated. (Module type cannot be specified with this format.)

DEFAULT_MODULE directive

Sets the module attributes of all Housecode|Units which are not defined in an alias directive. If not otherwise specified by this directive, the default module is the standard X-10 plug-in lamp module (StdLM).

START_ENGINE directive

Many of Heyu´s features require the Heyu state engine daemon heyu_engine to be running. This directive instructs Heyu how heyu_engine is to be started. With the default value of MANUAL, the engine must be started by entering ´heyu engine´ at the command line. With the value AUTO, the engine will be started automatically along with Heyu´s other background processes when ´heyu start´ is run. Example:
START_ENGINE AUTO

LOG_DIR directive

Use this directive to specify the directory in which the Heyu state engine daemon should write its log file ´heyu.log.<tty>´. The keyword ´NONE´ (which is the default) instructs Heyu to not write a log file.

Example:
LOG_DIR /var/log/heyu/

The log file will contain entries like appear in the Heyu monitor, and in addition, an entry whenever a script (excluding heyuhelper) is launched. It will also collect the text output of a launched script, if that output isn't redirected to a different file.

Two or more instances of Heyu running on the same computer can share a common log file by appending the keyword "COMMON". If each instance of Heyu is started with the base subdirectory switch (-0 through -9), each entry in the log file ´heyu.log.common´ will display the subdirectory number (0 through 9) from whence the signal originated.

Example:
LOG_DIR /var/log/heyu common

Note that the log file will continue to grow. Manually delete or trim it from time to time, or configure a Unix utility like ´logrotate´ to manage it.

DATE_FORMAT directive

Specifies how a numeric date is displayed. The syntax for this directive is:
DATE_FORMAT <order> [<separator>]
where <order> is
YMD => Year/Month/Day (Default)
DMY => Day/Month/Year

MDY => Month/Day/Year
and where the optional <separator> is ´/´ (Default), ´-´, or ´.´

Examples:
DATE_FORMAT YMD ´/´ => 2008/01/23 (Heyu default)
DATE_FORMAT YMD ´-´ => 2008-01-23 (ISO 8601 standard)
DATE_FORMAT DMY ´-´ => 23-01-2008 (many European countries)
DATE_FORMAT DMY ´.´ => 23.01.2008 (Germany traditional)
DATE_FORMAT MDY ´/´ => 01/23/2008 (USA traditional)

The specified order of Month and Day also applies to the date ranges specified for Timers in schedule files to be uploaded to the CM11A EEPROM. The separator between month and day must be ´/´, ´-´ or ´.´ but does not have to be the same as that specified with DATE_FORMAT. The separator between the begin and end dates may optionally be either ´-´ or ´:´. (Use the latter for clarity when the Month/Day separator is a ´-´.)
Examples:
TIMER smtwtfs 01/23-12/31 ... (with DATE_FORMAT YMD or MDY)
TIMER smtwtfs 23-01:31-12 ... (with DATE_FORMAT DMY)

Dates where the month name is spelled out are not affected by this directive and usually appear like:
Wed 23 Jan 2008

LOGDATE_YEAR directive

Instructs Heyu whether or not the dates reported for entries in the Log file and Monitor should include the year. The choices are YES or NO, with the default being NO, to omit the year.

TAILPATH directive

Use this directive to specify the full pathname of the system ´tail´ command if it´s not on the normal PATH accessable to Heyu. The Heyu command ´logtail´ will use this pathspec to call ´tail´.

HEYU_UMASK directive

Governs the permissions for files created by Heyu. The default for this directive is 0000 which results in files having permissions rw_rw_rw_. The value 0002 results in files having permissions rw_rw_r__ ; the value 0022 results in files having permissions rw_r__r__.
Example:
HEYU_UMASK 0002

USERSYN directive
SCENE directive

These directives, new to Heyu version 2, specify a semicolon-separated list of Heyu commands. These can then be executed sequentially by issuing only the usersyn or scene label as a Heyu command. In addition, the label can be used in a macro as if it were a command.

In the current release, the ONLY difference between a SCENE and USERSYN is whether they appear in the scene or usersyn list displayed by the ´heyu show´ command. What applies below for a SCENE applies equally to a USERSYN.

The format for either a scene or usersyn is similar:
SCENE Label command1; command2; command3; ...
USERSYN Label command1; command2; command3; ...

The commands in a scene can accept either actual device addresses or aliases. (The alias directive must appear before any scene directive which references it.)
Examples:

```
SCENE  dim_all  dim A1-3  8; dim B2  8; dim c7  8
SCENE  night_lights  off porch_light; on garage_light; dim hall 8
```

Issuing the command ´heyu dim_all´ would result in each of the commands defined in that scene to be executed in turn.

Scenes can also be defined with positional parameters which are replaced by actual parameters when they are executed. The positional parameters are designated by a ´$´ sign followed by a number 1 to N optionally enclosed in curly brackets, e.g., ´$2´, ´${7}´. The number represents the position of the parameter in the argument list supplied when the scene is executed.. A positional parameter can be used as part or all of either a device address or a dim level, but not as part of a scene label or a Heyu command

Examples:

```
SCENE  blinker  on $1; off $1; on $2; off $2
SCENE  Dim_all  dim A1-3  $1; dim B2  $1; dim c7  $1
```

These could then be executed by running:

```
heyu  blinker  A3  B7
heyu  Dim_all 7
```

Rules for scenes (interpret "scene" as meaning either scene or usersyn):

1. A scene label is a case-sensistive ASCII string (maximum length 32 characters) not beginning with ´-´ or ´_´ and not containing blanks or the ´+´ or ´$´ characters.

2. A scene label may not be the same as a Heyu command or any of its synonyms. A scene label may not duplicate a usersyn label and vice-versa. As of the current implementation, a scene label MAY be the same as an alias label, and it MAY be the same as a macro label, but these feature may be restricted in future releases if too much user confusion results.

3. Only "direct" and CM17A commands can be used in scenes - "administrative" commands like ´help´, ´info´, etc., are invalid in scenes, as are commands identified as "legacy" commands from Heyu version 1. (Running ´heyu help´ will identify each of these types of commands.)

4. A command in a scene may not be another scene.

5. Commands in scenes are checked for syntax when the configuration file is read, but there is no checking of any parameter containing a positional parameter until the scene is actually executed or specified in a macro. The syntax checking is performed as if the scene is to be executed at the command line, so if a scene contains a command which is invalid for an uploaded macro, that error won´t be flagged until the ´heyu upload [check]´ command is run.

6.There is no restriction on how a positional parameter is used within a parameter so long as the result is valid when the scene is executed. Use the optional curly brackets if the result could otherwise be ambiguous. The positional parameter is first replaced by the actual parameter, then the result is checked against the list of aliases.

7. The number of positional parameters supplied when a scene is executed must agree exactly with the number specified in the scene definition. Thus if ´$4´ is the highest numbered positional parameter specified in a scene, then 4 and only 4 parameters must be supplied when the scene is executed.

8. If more than one scene is used in a macro, or if scenes are intermixed with regular macro commands, they must be separated by semicolons.

9. Memory for scenes is dynamically allocated so there is no fixed size limit for scenes intended for execution from the command line. Bear in mind however the limited (1024 byte) EEPROM space in the CM11A when scenes are expanded in macros, although most common X10 commands occupy only 3 or 4 bytes when converted to binary.

10. The highest numbered positional parameter in a scene is by default limited to 8. (In most cases anything

over that is more likely an error.) But this limit may be changed with configuration directive MAX_PPARMS.

MAX_PPARMS directive

Specifies the highest numbered positional parameter allowed in a scene. The number may have any value between 1 and 999. The default is 8.

STATUS_TIMEOUT directive

Specifies how long Heyu will wait for a module to reply to a status command before timing out. The default is 2 seconds. Some modules, SwitchLinc dimmers in particular, may require increasing this to 3. The allowed limits for this directive are 1-5 seconds. (Don't use a value any higher than the minimum needed for satisfactory status reporting.) Example:

STATUS_TIMEOUT 2

SPF_TIMEOUT directive

Specifies how long Heyu will wait for a module to reply to a "SPecial Function" status command before timing out. The default is 3 seconds. To date, there is only one special function in Heyu, the display of the temperature as encoded in a preset command returned from a two-way thermostat or remote thermometer. (See below.)

RCS_DECODE directive
RCS_TEMPERATURE directive

This is the same directive - use either but not both. The name has been changed since Heyu now supports decoding other RCS status reports such as fan status on/off in addition to temperature.

This directive instructs the Heyu monitor to decode and display the status reports encoded in a Preset command transmitted from a two-way thermostat or remote thermometer employing the algorithm used by the RCS TX15-B and TXB16 thermostats (shown as a table in the thermostat user´s guide).

The Smarthome TempLinc Model 1625 remote thermometer uses the same algorithm for the temperature report, which is:

temperature = -60 + 32*(unit - 11) + (preset_level - 1)

If the Heyu State Engine is running, the decoded temperature is stored at the (fictitious) unit 0 address H0, where ´H´ is the housecode of the thermostat or thermometer. It can be recovered with the ´heyu rcstemp H´ command. Provided a valid temperature has been stored, any script launched by Heyu can recover the temperature as the value of environment variable X10_H0 or the environment variable alias for H0. (Since these environment variables are created only if valid data has been stored, a check for their existance, e.g., ["X10_H0" != ""], may be required before using them in a script or shell command.)

*** WARNING - WARNING - WARNING ***

The use of Heyu or any other power-line protocol software to control a heater lacking a built-in failsafe mechanism is not only unwise, it is hazardous to life and downright foolhardy. DON´T DO IT!

This directive may have the values: OFF for no decoding; a list if housecodes enclosed in square [] brackets for which decoding is to be performed; or ALL to decode any housecode. The default is OFF.

Examples:

```
RCS_DECODE  OFF      (no decoding)
RCS_DECODE  [CFH]    (decode on housecodes C, F, and H)
RCS_DECODE  ALL      (decode on any housecode)
```

Note: This directive need not be used in order to query the thermostat with the ´heyu rcs_req ...´ or ´heyu temp_req ...´ commands - it only enables the temperature or other status to be displayed in Heyu´s monitor, regardless of whether or not the thermostat´s transmission was initiated by these commands. (The

TempLinc Model 1625 remote thermometer can be programmed to transmit a temperature report any time the temperature changes.)

ACK_HAILS directive

Setting this directive to YES will instruct the Heyu state engine daemon (if running) to send a hail_ack with the default housecode whenever it receives a hail signal over the power line. The hail_ack is sent as if from a launched script. The choices are YES or NO, with the default being NO.
Example:
ACK_HAILS NO

AUTOFETCH directive

When a state command which returns the addressed state of a module is executed at the command line, Heyu by default (AUTOFETCH YES) instructs the state engine to first update the state file, since Heyu only automatically updates this file following an X10 function. Setting the value of the directive to NO disables this action for (only) those specific state commands. Most users will want to accept the default value of YES. See the description of the ´fetchstate´ command in man page heyu(1) for a more detailed discussion of this issue.

TIMER_LOOPCOUNT directive

Specifies the base loopcount for the fast timing loops needed by some CM17A and experimental commands. To determine the value for this directive, run ´heyu utility calibrate´.

FORCE_ADDR directive

Heyu doesn´t normally send a separate Housecode|Unit address byte for commands like all_lights_on which don´t actually require the unit number (the housecode is included with the function), or for extended commands which contain the unit number within the function code. However X-10´s ActiveHome software always sends this byte, generally using unit 13. This directives forces Heyu to always send an address byte. If always sending an address byte cures some problem you are having, choose YES. Otherwise take the default of NO.
Example:
FORCE_ADDR NO

SPOOLFILE_MAX directive

This directive sets the limit on the size of the spoolfile beyond which the Heyu relay daemon will rewind it once it detects a period of inactivity (currently 5 seconds). The size set by this directive may be between 20 and an absolute maximum 1048576 bytes (but don´t use this max value). The default limit is 1000000 bytes. For changes in this directive to become effective when Heyu is already running, use the ´heyu restart´ command.

Note that re-synchronization with the Heyu state engine daemon or monitor takes a couple of seconds, during which time Heyu will ignore incoming X10 signals or macro executions. So don’t set a low limit except for testing purposes.

SECTION directive

This directive and anything which follows it on the line is treated the same as a comment and totally ignored by Heyu. It is provided to enable user-definable breakpoints for use by an external program or script to reorganize the Heyu configuration file.

CHECK_RI_LINE directive

When the CM11A receives an X10 signal over the power line it asserts the Ring Indicator (RI) serial line for a short time prior to sending the signal to the computer. This directive tells Heyu whether or not to

check this serial line before attempting to transmit a command so as to help avoid collisions on the power-line. The default is YES. This check is normally bypassed for serial port hardware which does not support the serial control lines, so setting this directive to NO should only be required in exceptional situations. One such situation occurs with USB-Serial adapters using an older Prolific chip under Linux, resulting in the message "RI serial line may be stuck" after a long delay. Note that this check will be ineffective if the CM11A RI line is disabled by the following directive.

RING_CTRL directive

Setting the value of this directive to DISABLE instructs Heyu to disable the CM11A Ring Indicator (RI) line of the CM11A, which would otherwise be asserted each time an X10 signal is received over the power-line. In addition, Heyu commands like ´heyu ping´ and ´heyu wait´ which normally employ the CM11A Ring_Enable command for their functionality will substitute the CM11A Ring_Disable command. This directive is not totally effective because the CM11A asserts the RI line when it is powered up, before Heyu has a chance to disable it, e.g., in the event of a momentary interruption in AC power.

The ONLY reason for disabling the RI line is that some computers have a "Start on Ring" feature and/or "Awake from Sleep on Ring" feature which cannot easily be disabled in the computer´s BIOS and disabling the CM11A RI line prevents undesired startup of the computer. Otherwise the default value of ENABLE for this directive should not be overridden since the function to check the RI line before sending would then be ineffective.

For changes in this directive to become effective when Heyu is already running, use ´heyu restart´ to incorporate the change in the heyu_relay and heyu_aux daemons.

SEND_RETRIES directive

Heyu by default attempts to send an X10 address or function command to the CM11A up to 3 times, which is the default for this directive. If collisions or noise on the power line result in _occasional_ "Unable to send ..." messages, use this directive to bump up the number a little. Note: This directive will not help if the failure to send a command is due to a communication problem with the serial port.
Example:

```
SEND_RETRIES 5
```

POWERFAIL_UPDATE directive

When AC power resumes after an interruption, the CM11A will poll the computer with a constant stream of requests at one-second intervals for a time update and will accept no other command until the request is satisfied. The Heyu relay daemon by default automatically responds and updates the CM11A clock. The choices for this directive are either YES or NO, with the default being YES. Choosing NO for this directive instructs heyu_relay to respond with a truncated update which satisfies the CM11A but doesn´t actually reset the clock. If Heyu is already running, use ´heyu restart´ for changes in this directive to become effective in the heyu_relay daemon.

This directive allows the user to determine the accuracy of the CM11A clock when operating solely on battery power. E.g., run ´heyu setclock´ to set the CM11A clock to the system clock, unplug the CM11A from the wall socket for several hours or a day, plug it back in and run ´heyu readclock´ to compare the CM11A and system clocks.

NEWFORMAT flag (obsolete)

The word NEWFORMAT formerly caused the output displayed in the Heyu monitor to appear in a slightly different format. In Heyu version 2 the monitor output is very different and the older formats are no longer supported. This directive will be ignored if it appears in the configuration file.

==========

The following Directives are applicable when scripts are to be executed by Heyu in response to an X10 signal. (See ´man x10scripts(2)´ for details.)
==========

SCRIPT_MODE directive

The options are HEYUHELPER or SCRIPTS, with the default being SCRIPTS.

If HEYUHELPER is chosen, a shell script named ´heyuhelper´ on the user´s path is executed every time an X10 function is received by Heyu from the interface. The argument supplied to the script is of the form <housecode|last_unit|function>, where:
housecode is the lower case letter a-p associated with the function
which resulted in execution of ´heyuhelper´
last_unit is the last unit code received on that housecode.
function will be one of the following: On, Off, AllOn, AllOff,
Dim, Bright, LightsOn, LightsOff, Preset, Extended, Hail, HailAck,
Status, StatusOn, StatusOff, DataXfer.

Examples of heyuhelper arguments: a1On b3Off c14Preset

Note: The AllOn command in Heyu version 1.xx was interpreted to mean LightsOn (All Lights On). It is not a native CM11A command. In Heyu version 2, it instead indicates the simple ´On´ function whenever all units 1-16 are addressed.

Another difference from Heyu version 1.35 is that the heyuhelper script is not launched when an uploaded macro is executed.

The heyuhelper script is only executed when X10 functions are received from the interface - not when addresses or powerfail messages are received or when functions are sent.

All other script-oriented directives with the exception of SCRIPT_SHELL and SCRIPT_CTRL are ignored with the HEYUHELPER option. Choosing SCRIPTS allows one to use the full features of Heyu scripting (including execution of an existing ´heyuhelper´ script).

SCRIPT_CTRL directive

Globally enables or disables launching of scripts by Heyu. The default value of ENABLE permits scripts to be launched; the value DISABLE inhibits them from being launched. This will hopefully be useful when the configuration file contains many SCRIPT directives and it is desired to temporarily disable them without having to delete them or comment them all out. All scripts are affected, including heyuhelper and powerfail or rfflood scripts. (This directive may be overridden at the command line by the ´heyu script_ctrl enable|disable´ command.)

SCRIPT directive

Specifies the Linux/Unix command to be launched and the conditions under which it will be launched. The conditions for launching the command may also (or additionally) be specified separately in a LAUNCHER directive.
The format is as follows:
SCRIPT [-l <label>] <launch conditions> :: [exec options] <command line>

The script label is optional as long as the all the launch conditions are contained within the SCRIPT line. It can be any arbitrary case-sensitive name up to 32 characters in length, not beginning with a ´-´ character. If no label is specified, Heyu will create one using the line number where it appears in the config file to make it unique, e.g., ´Script_21´.
If the launch conditions are wholly or in part specified with a LAUNCHER directive, a script label is mandatory, so that Heyu can match up the script with the assocated launcher.

Launch conditions can be quite simple, like ´A1 on´, or quite complex. In addition, there are special launch condition for extraordinary situations: −powerfail´, ´-rfflood´, ´-timeout´.
See x10scripts(5) for a full description of launch conditions.

Keep in mind that Heyu will normally launch a script only for the first set of launch conditions which are satisfied, i.e., only one script is launched per signal or special condition, although this restriction can be overridden. Launch conditions are tested in the order they appear in the configuration file.

The symbol ´::´ (two colons) is always required (so that Heyu can distinguish when the launch conditions end and the command line starts).

Exec options are zero or more of the following:
-xtend (or simply -x) Create an environment compatible with Xtend scripts.
(No other options are compatible with -xtend.)

-rawlevel (or simply -r) Use native raw dimlevels instead of percentages
for state bitmaps in the Heyu environment.

-noenv (or simply -n) Provide no additional environment variables
beyond those already in the user´s environment.

The <command line> can be any executable binary, shell script, or shell command.

Examples:
```
SCRIPT  D1 on ::  doorbell_actions.sh
SCRIPT  -l wake_em_up  C1 on :: play $HOME/music/reveille.wav
SCRIPT  -l PowerFail -powerfail :: echo "There was a power failure" | mail
SCRIPT  -l RF_Flood  -rfflood :: heyu on Siren
SCRIPT  -l Timeout -timeout timer2 :: echo "Timeout on Timer number 2" | mail
```

If a script directive references an alias in place of a housecode|unitcode, the alias directive must have been defined earlier in the configuration file.

LAUNCHER directive

This optional directive can wholly or in part specify the condition or conditions under which the associated script will be executed (or ´launched´).
The format is:
LAUNCHER <script label> <launch conditions>; <launch conditions>; ...
where:
<script label> identifies the associated script with the same label.

See x10scripts(5) for a full description of what can appear in <launch conditions>.

Multiple launchers for the same script may be continued on the same line following a semicolon, or may be configured as separate LAUNCHER lines.

Example:
If no launch condition was specified in the ´wake_em_up´ script above, it would be specified in a launcher like this:
```
LAUNCHER  wake_em_up  C1  on
```

LAUNCHER_SCANMODE directive

See section "SCAN MODE" in man page x10scripts(5) for details.

SCRIPT_SHELL directive

Defines the shell used when a script is executed. (Provide the full pathname.) The default is to use the shell defined by the SHELL variable in the user´s environment, or /bin/sh if no such variable exist.

LAUNCHPATH_PREFIX directive
LAUNCHPATH_APPEND directive

Prefix and/or append an additional directory or directories to the existing $PATH in the environment provided for Heyu scripts. (This works only for shells where the path separator is a semicolon.)
Example:

```
LAUNCHPATH_APPEND  /etc/heyu/scripts
```

LAUNCH_SOURCE directive

Sets the default source or sources of the X10 signal allowed to launch a script as part of the <launch conditions>. One or more of the following may be specified. The default is RCVI.

SNDC Sent from the Heyu command line.
SNDM Sent upon execution of an uploaded macro by a Timer.
SNDT Sent upon execution of an uploaded macro by a Trigger
SNDP Sent by Heyu from a power-fail script.
RCVI Received over the power line via the CM11A interface.
RCVT A Trigger which initiates execution of an uploaded macro.
SNDA Sent over the power line by the heyu_aux daemon.
RCVA Virtual data received from the heyu_aux daemon.
ANYSRC All of the above.
NOSRC None of the above (sources must be specified in the <launch conditions>).

Example:

```
LAUNCH_SOURCE  rcvi  sndc
```

ISDARK_OFFSET directive

Defines the offset in minutes from Dusk and Dawn to use for the script environment logical variable ´X10_isDarkTime´. Its default value of 30 minutes means that ´X10_isDarkTime´ will be TRUE from 30 minutes after Dusk until 30 minutes before Dawn. This directive also determines the interval during which the launch condition flag ´dark´ is TRUE.

ENV_ALIAS_PREFIX directive

Determines whether the prefix of ALIAS environmental variables provided for scripts is upper or lower case, i.e., ´X10_´ or ´x10_´. The choices are LC or UC, with the default being LC for lower case.

This directive enables compatibility with scripts written for earlier releases of Heyu (which used the upper-case prefix), however its use is discouraged - please modify your scripts to use the lower-case ´x10_´as soon as practicable.
Note that this prefix applies _only_ to ALIAS environment variables - all other Heyu environment variables use the upper-case prefix. E.g, if the configuration file defines:

```
ALIAS  porch_light  A6
```

then the environment variables provided will be:

```
x10_porch_light=nnnn
X10_A6=nnnn
```

STATE_FORMAT directive

If set to the value OLD, the output of the state commands ´onstate´, ´dimstate´, ´chgstate´, and ´addrstate´ revert to the former heyuhelper style, e.g., a2Dim. Taking the default value NEW displays output as either 1 (True) or 0 (False) when a housecode|unit parameter is provided, or a unit bitmap of the state (bit 0 -> unit 1) as a decimal integer when the parameter is a housecode only. This directive is intended to provide compatibility with existing scripts.

STATE_CTRL directive

With the default value of "SINGLE" for this directive, the output of state commands ´onstate´, ´dimstate´, ´chgstate´, and ´addrstate´ is boolian 0 or 1 when the address parameter includes the unit, or a bitmap if the address parameter is only a housecode. Specifying the value of this directive as "BITMAP" results in the output being a bitmap regardless of whether or not the address include a unit.

==========

The following Directives are applicable when a schedule of Timers and Macros is to be uploaded by Heyu to the CM11A´s EEPROM.

==========

SCHEDULE_FILE directive

Instructs Heyu to use the named file as the user´s schedule file (which must be in the same directory as the configuration file). The default is x10.sched.

Example:

```
SCHEDULE_FILE  vacation.sched
```

MODE directive

Heyu can operate in one of two modes insofar as uploaded timers and macros are concerned. But before getting into these, first a few words about the CM11A interface´s internal clock:

The CM11A clock has no notion of actual dates, but is merely a day counter which starts from 0 and increments daily until it reaches 365 (a total of 366 days), after which it rolls over to 0. An independant counter keeps track of the weekday, starting at 0 (Sunday) and incrementing daily through 6 (Saturday) before rolling over to 0. Additional counters track hours, minutes, and seconds with the usual rollovers. (The CM11A has no notion of Standard versus Daylight Time.) The day and weekday counters increment when the time rolls over to 0:00:00.

In either mode, the CM11A clock is maintained by Heyu on Standard Time thoughout the year. Timers scheduled by the user in Civil (i.e. wall clock) time are automatically divided by Heyu into separate timers for periods of Standard and Daylight Time during the year. Time management is intended to be handled by Heyu transparently to the user.

In COMPATIBLE mode, the CM11A clock is configured such that Day 0 corresponds to Jan 1st in the current year and the uploaded schedule of timers and macros is prepared to run for 366 days, i.e., through either Dec 31st or the following Jan 1st, depending on whether the current year is a leap or common year. In order to maintain the proper correspondence, the user´s schedule must be re-uploaded on Jan 1st, at least in years following common years. (The CM11A will otherwise keep chugging away, but events fired off will be a day in error.)

In HEYU mode, the CM11A clock is configured such that Day 0 corresponds not to Jan 1st but to today´s date. The uploaded schedule can be configured to run for any period between 1 and 366 days via the PROGRAM_DAYS directive. (See the description of that directive for reasons for wanting this.) The user´s schedule can be re-uploaded at any time prior to expiration of the PROGRAM_DAYS period and correspondance will be maintained for the next period of PROGRAM_DAYS. Note however that if this period is allowed to expire without re-uploading the schedule, the CM11A will cease to fire off any programmed events (unless of course the period is set for 366 days).

The schedule should be programmed for the entire year regardless of the MODE or PROGRAM_DAYS

directives. Based on those directives, Heyu determines what parts of it to use and what parts to ignore. In HEYU mode, events scheduled for execution on dates earlier than today´s date wrap around into next year. Examples:

```
MODE  COMPATIBLE
MODE  HEYU
```

*** WARNING - WARNING - WARNING ***

Users running PCs configured for dual-boot between Linux and MS-Windows should NOT operate Heyu in HEYU mode if they expect to ever manually execute X-10´s ActiveHome(TM) program (or allow its "Communications Bridge" driver to be launched at boot time). Here´s why:

If the CM11A detects that its AC power has been interrupted, even for a very short time, it will continually issue commands back to the PC requesting a clock update. ActiveHome and/or its Communications Bridge will _silently_ comply with this request (as will Heyu´s Relay), but ActiveHome will set the CM11A clock to an incorrect date and time for HEYU mode. (This is easily and automatically corrected by running the ´heyu setclock´ command, but the PC has to be rebooted back into Linux/Unix to do it.)

PROGRAM_DAYS directive

When operating Heyu to upload timers and macros to the CM11A interface in HEYU mode, this directive instructs Heyu to compile the uploaded schedule to run for the number of days specified (1-366), beginning today. This directive is ignored when Heyu is operated in COMPATIBLE mode. The default value is 366 days. Example:

```
PROGRAM_DAYS  90
```

The advantage to running for a shorter number of days becomes evident when timers are programmed for Dawn- or Dusk-relative times. Because of the limited size (1024 bytes) of the CM11A EEPROM memory, Heyu uses a stair-step approximation to the daily times of Dawn and Dusk over the PROGRAM_DAYS interval. The unequal-length step intervals are determined iteratively to minimize the error while fitting in the available EEPROM memory space.

As a result, even a single Dawn- or Dusk-relative timer will generally occupy most of the EEPROM space. However as more timers are added to the schedule, the stair-steps are recalculated so everything will fit, although at the expense of a larger error.

LATITUDE directive
LONGITUDE directive

These are used to allow Heyu to compute the times of dawn and dusk over the year for your particular location. There are no defaults.

The format of the directive line is the word LATITUDE followed by a space and then the direction, degrees, a colon and the minutes. The LONGITUDE directive follows the same format. Example:

```
LATITUDE     N37:41
LONGITUDE    W121:52
```

Note: Versions of Heyu earlier than 2 did not use a direction (N/S/E/W) symbol and assumed the convention that positive Latitude is North and positive Longitude is West. For compatibility, Heyu version 2 will make the same assumptions if the direction symbol is omitted.

COMBINE_EVENTS directive

This directive instructs Heyu whether or not to combine events which have the same day-of-week code, date range, type (i.e., Clock, Dawn, or Dusk based), and time-of-day. Heyu then needs to upload only a single timer executing a concatenated macro, which saves considerable EEPROM space when Dawn/Dusk

relative timers are programmed.
The choices are YES or NO, with the default being YES.
The downside to using this feature is that the Heyu-assigned name of the concatenated macro will not be easily recognizable when displayed in Heyu´s monitor.

COMPRESS_MACROS directive

This directive instructs Heyu to merge unit codes for macro elements which have the same command, housecode, and "data" (e.g. dim level), and also eliminate duplicate elements. It can save a few bytes of EEPROM space, when macros with similar elements are concatenated.
The choices are YES or NO, with the default being NO.

Example:
```
  macro mac1 0  on A1; on B1; on A2; on B3
```
becomes:
```
  macro mac1 0  on A1,2; on B1,3
```

Warning: If you program a macro like:
```
  macro mac1 0  on A1; off A1; on A1; off A1; on A1; off A1; ...
```
and expect to have a lamp on A1 repeatedly blinking on and off, better set this directive to NO, else the macro will be compressed to a single on and off. (Heyu maintains the ordering in macros to the extent possible.)

DAWNDUSK_DEF directive

By default Heyu defines Dawn and Dusk to be Sunrise and Sunset. This directive allows globally defining them instead as the morning and evening times of several standard twilights, as follows:
RISESET (or simply ´R´) - Sunrise and Sunset (default)
CIVIL (or simply ´C´) - Civil Twilight
NAUTICAL (or simply ´N´) - Nautical Twilight
ASTRONOMICAL (or simply ´A´) - Astronomical Twilight

Example:
```
 DAWNDUSK_DEF  CIVIL
 DAWNDUSK_DEF  N
```

DAWN_OPTION directive
DUSK_OPTION directive

These directives instruct Heyu how to assign the time for Dawn or Dusk in each timer subinterval. The options for this directive are:
FIRST - Use the Dawn/Dusk time for the first day in the subinterval.
EARLIEST - Use the earliest of the times for any day in the subinterval.
LATEST - Use the latest of the times for any day in the subinterval.
AVERAGE - Use the arithmetic average of the times in the subinterval.
MEDIAN - Use a time halfway between the earliest and latest times.

The default is FIRST, which is the most convenient for comparing Heyu´s computations with Dawn/Dusk times published in newspapers or by the US Naval Observatory. (Remember that Heyu´s times are Standard Time, while newspapers generally publish Civil [wall clock] times.)

MIN_DAWN directive
MAX_DAWN directive
MIN_DUSK directive

MAX_DUSK directive

These directives allow bounds to be placed on the times of Dawn and Dusk computed by Heyu. For example, setting the value for MIN_DAWN to 06:30 will instruct Heyu to execute a timed event scheduled for dawn at 06:30 instead, whenever the computed time for dawn is earlier than 06:30. These directives may be useful for users in extreme latitudes, when for example a lamp is scheduled to turn on at dusk and turn off at 9:30 PM. But if during the summer months dusk actually occurs after that hour, then the on/off cycle of the lamp would normally be reversed, turning on at that late dusk and remaining on until 9:30 PM the following day.

The value for these directives are specified as hh:mm Civil (i.e., wall-clock) time. The directives may be disabled with the word OFF in place of a time. The default is OFF.

Heyu version 2 now has timer options for conditional compilation of events depending on the values of Dawn and Dusk, so these directives may no longer be useful.

(Note: Due to Heyu´s optimizations, the order in which events scheduled for the same time are actually executed is not necessarily that in which they appear in the user´s schedule file. Allow at least a one minute time difference when order is important. The actual order can be determined from the list of timers in the OUTPUT TIMERS section of the report.txt file written when a schedule is uploaded or upload checked.)

DAWN_SUBSTITUTE directive
DUSK_SUBSTITUTE directive

These directives are obsolete and will be ignored.

REPORT_PATH directive

Reports ´report.txt´ and/or ´cronreport.txt´ created when the command ´heyu upload [check|croncheck]´ is run are by default written in the Heyu base directory, i.e., the directory where the configuration file is stored. This directive instructs Heyu where the user would like them written instead. The full pathspec is required (127 characters maximum).
Example:

```
REPORT_PATH ./
```

WRITE_CHECK_FILES directive

When a schedule is actually uploaded to the CM11A´s EEPROM, Heyu records critical information in the files ´x10record´, ´x10macroxref´, and ´x10image´. For debugging or informational purposes, this directive instructs Heyu to write these files with a .check extension when ´heyu upload check´ is run.

In addition, this directive instructs Heyu to write ´x10image.hex´ which is a human-readable hex dump of the EEPROM memory image.

The choices here are YES or NO. The default is NO.

REPL_DELAYED_MACROS directive

This directive instructs Heyu to replace events having delayed macros with new events and new undelayed macros when possible. (The purpose is to avoid pending delayed macros, which are purged when a new schedule is uploaded.) The choices are YES or NO, with the default being YES.

ASIF_DATE directive

Used for test purposes, this directive instructs Heyu to substitute the argument date for today´s date when processing the user´s schedule file.
The date format is yyyymmdd.
Example:

ASIF_DATE 20020621

ASIF_TIME directive

Used mainly for test purposes, this directive instructs Heyu to use the argument time of day instead of the system time when processing the user´s schedule file. Its format is hh:mm (Civil Time). The default is to use the current system time.
Example (simulates operation at Noon):
ASIF_TIME 12:00

FEB_KLUGE directive

Heyu ver 2 always interprets 02/30 as the last day of February when used as a start or stop date. This directive instructs Heyu to also interpret 02/28 and 02/29 as the last day of February. The options here are YES or NO, with YES being the default. (If you choose NO, programming 02/28 or 02/29 in the wrong year could result in a skipped day, an event occurring twice, or possibly an error, depending on circumstances.)

RESOLVE_OVERLAP directive

When uploaded timers specify Dawn/Dusk periods with different begin and end points, Heyu resolves the situation such that the approximation time used for Dawn and/or Dusk is the same for every timer on any given day. There are two methods for doing this and in most cases the newer method appears to give better results in term of maximum error on Dawn or Dusk. This directive allows returning to the previous method if more satisfactory. The choices here are OLD or NEW, with the default being NEW.
If the timers in your schedule involve only clock times or don't involve both Dawn and Dusk-relative times, either choice will yield the same result.
(A future release of Heyu will make this determination automatically and obsolete this directive.)

FIX_STOPSTART_ERROR directive

With parameter "YES" is a work around for the CM11A EEPROM bug "Stop Time = Start Time" which is occasionally encountered when uploading specific schedules. It should be used (generally as a "config" line in the schedule file) only when needed as it results in less efficient use of the EEPROM memory space.

==========
The following directives are applicable to CM17A
==========

RF_POST_DELAY directive

This directive specifies the delay in milliseconds following each standard RF command to a CM17A "Firecracker" to allow time for the transceiver to respond with its power line signal. The default is 850 milliseconds.
Example:

RF_POST_DELAY 850

RF_FARB_DELAY directive
RF_FARW_DELAY directive

Specifies the delay in milliseconds following respectively a ´farb´ or ´farw´ RF command to a CM17A. The default is 850 milliseconds. This delay is specified separately from that for the standard RF commands so that these commands can be used to control devices that don´t involve a transceiver and power line signals.

RF_BURSTS directive

Specifies how many RF bursts (1-5) are to be transmitted with each actuation of the CM17A commands fon, foff, fdim, fbright, flightson, flightsoff, falloff, farb, and farw. Some transceivers will repeat some power line signals more than once when they receive the default 5 RF bursts. This is ordinarily not a problem except possibly when the particular power line signal is used to launch a script, in which event this directive can be useful.

This directive has the syntax:
´RF_BURSTS <command> <bursts> [<command> <bursts> [...]] ´, e.g.,
RF_BURSTS falloff 3 flightson 3 fdim 4

Note: Previous releases of Heyu required that each command be entered on a separate RF_BURSTS line and multiple RF_BURSTS directive lines were allowed.

The default for all CM17A commands is 5 bursts. For commands fdim, fbright, and farb where the total number of bursts is specified on the command line, the CM17A will be actuated multiple times, with the maximum number of bursts each time specified by this directive.

The RF_BURSTS directive is not used for the "fast" ffdim, ffbright, and ffarb commands.

TIMER_LOOPCOUNT directive

Specifies the base countdown for fast timing loops. These are usually needed to actuate the CM17A for the "fast" ´ffdim´, ´ffbright´, ´ffarb´, and ´ffarw´ commands. The value to use for this directive is obtained by running ´heyu utility calibrate´.

For the CM17A fast commands, the default value of zero instructs Heyu to use a kernel timer instead of a timing loop, but the resolution of the kernel timers for most Unix/Linux kernels is not fine enough for this purpose.

CM17A_BIT_DELAY directive

Specifies the programmed delay in _microseconds_ between bits sent to the CM17A device. X10´s RF protocol specifies a minimum of 500 microseconds and the default here is 500 microseconds. For Linux at least the minimum resolution of the kernel timer is usually much longer than this, typically 10 milliseconds, so changing the value probably won't accomplish anything except for the "fast" CM17A commands when a timing loop is used.

DISPLAY_RF_XMIT directive

Determines whether an entry will appear in the monitor and log file when a CM17A command is executed. The choices are NO or YES or VERBOSE, with the default being YES. Using VERBOSE is like YES but the triplet [XX YY B] is appended to each displayed entry. The XX, YY, and B are respectively the two hex bytes and decimal bursts which could be sent with the equivalent ´farb XX YY B´ command.

One quirk is that with multiple CM17A commands the RF entries will not necessarily be properly interleaved with the entries for transceived power line signals resulting from the RF command. This is because the CM11A doesn´t instantly report reception of the power line signals - there can be a delay of a second or more. The only workaround is to increase the RF_POST_DELAY to 2000 or 3000 milliseconds, which really slows down the RF commands.

RF_NOSWITCH directive

This directive determines whether actuation of the built-in appliance module switch on an X-10 TM751 or RR501 is to be suppressed when the transceiver receives the fon, foff, or falloff RF signal on the unit code for the switch. Setting the parameter of this directive to YES will suppress actuation of the switch with its accompanying loud audible click. The default is NO.

==========

The following directives provide controls for RF input to Heyu from the auxiliary port daemon heyu_aux. They´re more fully explained in man page x10aux(5).

==========

TTY_AUX directive

Specify the auxiliary serial port and RF receiver device.

TTY_AUX <serial port> <receiver device>
where <receiver device> is W800RF32A, MR26A, or RFXCOM. Example:
TTY_AUX /dev/ttyS1 W800RF32A

There is no default for this directive.

SUPPRESS_RFXJAM directive

Older firmware versions of the RFXCOM receiver sent a special "jamming" signal when they detected RF jamming, but the system was prone to many false positives and the feature was removed. This directive allows suppressing the jamming signals from older RFXCOM versions by treating them as RF Noise.
The options for this directive are YES or NO, with the default being NO.

ALIAS directive

For RF signals received by the W800RF32A from security remotes and sensors, this ALIAS format maps the device to a housecode|unit address.
ALIAS <label> <housecode|unit> <device model> <ID> [<ID> [<ID>...]]
where <ID> represents the security ID of a device expressed as a hexadecimal number, either with or without the "0x" prefix. Up to 16 security IDs can be associated with a single housecode|unit address in the case of multiple security remotes of the same model. (Each sensor device should be mapped to its own address.) Examples:
ALIAS kitchen_door F11 DS10A 0x3c
ALIAS keychain_remote G8 KR10A 2a b4 c7

The ID for entertainment remotes like the UR81A is fixed and included in the model, so isn't specified with the ALIAS directive.

TRANSCEIVE directive
RFFORWARD directive

Determine which housecodes which are to be transceived to power line code (source SNDA) by the heyu_aux daemon and which housecodes are forwarded directly (source RCVA) to the heyu_engine daemon for launching scripts or whatever. These applies only to signals from Standard X10 remotes, not security or entertainment remotes.
TRANSCEIVE <list>
RFFORWARD <list>
where <list> may be the keywords ALL or NONE, or may be a string of housecode enclosed in square [] brackets. Additionally, the keyword ALLEXCEPT followed by the housecode string in square brackets may be used. Example:
TRANSCEIVE [BFH]
RFFORWARD [DE]
which will transceive standard X10 RF signals on housecode B, F, and H, and forward X10 RF signals on housecodes D and E. RF signals on all other housecodes will be ignored. Another example:
TRANSCEIVE [BFH]
RFFORWARD ALLEXCEPT [BFH]

The default for the TRANSCEIVE directive is ALL, and for the RFFORWARD directive is NONE, meaning that signals on all housecodes will be transceived. The same housecode cannot be both transceived and forwarded.

Certain special module types have options to override the TRANSCEIVE and RFFORWARD lists for specific units and functions within a housecode.

TRANS_DIMLEVEL directive

This directive specifies the dim level (1-22) for each RF Dim or Bright signal transceived (or forwarded) by heyu_aux. The default level is 2 which results in a brightness level change of about 6 percent in standard X-10 Lamp Modules.
Example:

```
TRANS_DIMLEVEL  2
```

AUX_REPCOUNTS directive

This directive instructs heyu_aux how to handle multiple bursts in an uninterrupted sequence by providing 3 numbers:

```
AUX_REPCOUNTS  <MIN> <REPEAT> <MAX>
```

See man page x10aux(5) for details. Most users won't need to change the defaults (1 8 200) for this directive.

DISPLAY_RAW_RF directive

This directive instructs Heyu whether or not to display the raw RF data bytes from the receiver device. The choices are the default "NONE" to not display any raw data, "NOISE" to display data which heyu_aux judges to be RF noise, or "ALL" to display both noise and normal raw RF data. Only change from the default for testing or debugging.

SECURID_16 directive

This directive determines whether Heyu will use the full 16-bit ID of security sensors when received by a RFXCOM receiver in variable length packet mode. It is provided primarily for temporary backward compatibility in case the user has a large number of sensors configured with 8-bit IDs, until they can be reconfigured. The parameters are YES or NO, with the default being YES. The ID of a security device which transmits only an 8-bit ID is taken to be the lower byte of the 16-bit ID.

SECURID_PARITY directive

Security sensor transmissions include a parity check on the upper 8-bits of a 16-bit ID. Some sensors, notably the Aux channel of the DS90 Door/Window sensor, incorrectly report the parity on half of the RF repetitions. This directive with parameter YES (which is the default) instructs Heyu to check the parity. The parameter NO instructs Heyu to ignore the parity. Use NO only if you have a sensor exhibiting this bug.

HIDE_UNCHANGED directive

This directive applies only to transmitters identified by their module type as having a "heartbeat", i.e., sensors which periodically repeat their current state as an "I am alive" indication. The choices are YES or NO, with the default being NO. With YES, the sensor signal is displayed in the monitor and log file only when there´s a change from the previous state of the sensor or if the signal launches a script.

HIDE_UNCHANGED_INACTIVE directive

As above, but for the "func Inactive" pseudo_signal.

ORE_LOWBATTERY directive

Defines for Oregon sensors which transmit a battery level the percentage at or below which Heyu will raise the low-battery ("LoBat") flag. The default is 20%. Note that many Oregon sensor models transmit a low battery indicator rather than a battery level, and for these models this directive has no effect.
Example:
ORE_LOWBATTERY 10%

ORE_CHGBITS_T directive
ORE_CHGBITS_RH directive
ORE_CHGBITS_BP directive

Define for Oregon Temperature, Relative Humidity, and Barometric Pressure sensors the difference between succesive data values required for the data value to be identified as "changed". The parameter is the number of least counts for the data in question. The default for each is 1. See man page x10oregon(5) for details.

ELS_CHGBITS_CURR directive

Define for the Electrisave sensor the difference between successive current values required for the data to be identified as changed. The parameter is the number of least counts for the data. The default is 1, which corresponds to 0.1 Ampere. See man page x10oregon(5) for details.

OWL_CHGBITS_POWER directive
OWL_CHGBITS_ENERGY directive

Define for the Owl CM119 sensor the difference between successive Power and Energy values for the data to be identified as changed. The parameter is the number of least counts for the data in question. The default for each is 1, corresponding to 0.001kW and 0.0001kWh respectively. See man page x10oregon(5) for details.

OWL_DISPLAY_COUNT directive

Determines whether the raw data is displayed in the monitor/logfile for Owl CM119 sensors. The parameter choises are YES or NO, with the default being NO.

ORE_DATA_ENTRY

Defines whether data is entered in Oregon native or scaled units for Oregon sensor emulation. The parameter choices are NATIVE or SCALED, with the default being NATIVE. See section "OREGON SENSOR EMULATION" in man page x10oregon(5) for details.

ORE_DISPLAY_CHAN directive

Determines whether the channel number of an Oregon sensor is displayed in the logfile, monitor, and ´heyu show oregon´ table. Setting the parameter of this directive to NO suppresses display of the channel number, while with the default YES the channel number is displayed.

ORE_DISPLAY_FCAST

Determines whether the "weather forecast" transmitted by Oregon Barometric Pressure sensors (currently module types ORE_THB1 and ORE_THB2) is displayed in the logfile, monitor, and in ´heyu show oregon´. The choices are YES or NO, with the default being NO. The displayed forecast will be "Sunny", "PtCloudy", "Cloudy" or "Rain". The algorithm employed in the sensor firmware is currenly unknown but is assumed to be related only to the value of and changes in BP. (Indoor temperature and humidity data from a sensor rated only for indoor use can hardly have a bearing on a weather forecast.)

ORE_DISPLAY_COUNT directive

Determines whether the raw data is displayed in the monitor/logfile for Oregon Wind and Rain sensors. The parameter choises are YES or NO, with the default being NO.

RFXCOM_DISABLE directive

The RFXCOM RF receiver hardware may be configured to suppress signals from specific sensor types to improve noise immunity and/or to ignore foreign signals from nearby sensors. Its use supercedes the directive RFXCOM_ENABLE, which is now obsolete since newer RFXCOM firmware supports a wider variety of signal types.
The syntax is:
RFXCOM_DISABLE <type> [<type> [<type> ...]]]

The signal types which may be disabled are currently:
OREGON
ARCTECH
ATIWONDER
KOPPLA
X10 (Standard and Security X10)
HE_UK (HomeEasy - UK and KAKU protocols)
HE_EU (HomeEasy - EU protocol)
VISONIC (Receivable only by RFXCOM VISONIC receivers.)

All but HE_UK and HE_EU may be abbreviated to three characters.

Examples:
RFXCOM_DISABLE OREGON
RFXCOM_DISABLE ORE ARC ATI

RFXCOM_ENABLE directive

This is now obsolete. See RFXCOM_DISABLE above.

==========

The following directives provide controls for Heyu´s security system features.

==========

ARM_MAX_DELAY directive

When the ´heyu arm max ...´ command is entered at the command line or received from the RF auxiliary device, Heyu´s security flag is initially set to "armpending". At the end of the delay (in seconds) specified by this directive, the security flag is changed from "armpending" to "armed". The following example specifies the delay to be 30 seconds:
ARM_MAX_DELAY 30

The default for this directive is 0 (zero), which sets the "armed" flag immediately regardless of the "max" or "min" parameter.

ARM_REMOTE directive

This directive controls the behavior of Heyu when an ´arm´ signal is received from an RF Security remote (source RCVA). The allowed parameters are AUTO or MANUAL, with the default being AUTO. If set to MANUAL, the signal does not automatically set the global security flags, thus allowing the signal to be used to launch a script to customize the arming process, e.g., to first check that all doors and windows are closed and take an alternate action if they are not. This directive has no effect for ´arm´ commands originating from the command line or other sources.

INACTIVE_TIMEOUT directive

Most security sensors transmit a signal with their current state (known as a "heatbeat") every 60-90 minutes to let the host system know they are operational.
RFXSensors do the same thing every 40 or 80 minutes, depending on whether hardware configured for fast or slow sampling.
Oregon sensors transmit at intervals of 11 to 120 seconds, depending on model.

Heyu records the times the signals are received. The commands ´heyu sensorfault´ and ´heyu show sensors´ compare the elapsed times since the last received signals with the value specified with this directive to judge whether a sensor has become inactive.

The value for this directive may be specifies in seconds or in hh:mm:ss format. The default is 4:00:00 or 4 hours. Example:
INACTIVE_TIMEOUT 4:00:00

This directive (or its default) can be overridden for a specific sensor having a heartbeat by appending the parameters "IATO hh:mm:ss" to the ALIAS directive which maps that sensor to a Hu addresss.

INACTIVE_HANDLING directive.

The parameter choice is NEW or OLD, with the default being NEW. Choosing OLD reverts to the previous way Heyu handled the situation when a sensor having a heartbeat failed to send a signal within the interval defined by the INACTIVE_TIMEOUT directive. See man page x10scripts(5) for details.

ARM_LOGIC directive

The value of this directive may be specified as STRICT, LOOSE, or MEDIUM. The default is MEDIUM.

When specified as STRICT, the state of the security flags may be changed from "disarmed" to "armed" or "armpending" and vice-versa, but no changes between "armed" and "armpending" or between "home" and "away" are allowed. I.e., the system must be in the "disarmed" state before any ´heyu arm ...´ command can take effect.

When specified as LOOSE all changes are allowed. A ramification is that the ´arm max ...´ command when the system is already in the "armed" state will effectively disarm the system - put it in the "armpending" state - for the duration specified by the ARM_MAX_DELAY directive for as long as this command continues to be executed.

When specified as MEDIUM, all changes are allowed except the transition from "armed" to "armpending".

Example:
ARM_LOGIC MEDIUM

==========

The following experimental directives are otherwise undocumented and may or may not be useful for testing, hacking, or troubleshooting purposes.

==========

CM11A_QUERY_DELAY directive

On some systems a short delay appears to be required between a message from the CM11A that it has incoming powerline data and Heyu´s polling for that data. The parameter for this directive specifies the delay time in milliseconds. The default is 0, indicating no delay. Use of this directive may resolve the problem when Heyu reports: "The interface didn’t answer a getinfo response." when attempting to receive incoming powerline data.

Example:
CM11A_QUERY_DELAY 10 # 10 millisecond delay

LOCK_TIMEOUT directive

Before a Heyu process can write to the spool file it must first acquire the writefile lock. This directive specifies how long a Heyu process will spend trying to acquire the lock before timing out and quitting. Having a number of Heyu processes active at the same time and processes which hog the spool file, e.g., when sending commands to the CM11, may require increasing this time. The parameter for this directive is the time in seconds to spend, with the default being 10 seconds.
Example:
LOCK_TIMEOUT 20

DISPLAY_SENSOR_INTV directive

Setting the parameter of this directive to YES instructs Heyu to append to monitor/logfile entries for an Oregon or Security sensor the elapsed time interval in hh:mm:ss since the previous transmission from that sensor.

LOGDATE_UNIX directive

This directive can be used to replace the normal date/time string displayed in the logfile and monitor with Unix time, i.e., seconds from the 1 Jan 1970 epoch, to millisecond precision. The parameter for this directive is YES or NO, with the default being NO.

ORE_ID_16 directive.

Setting the parameter of this directive to YES expands the ID code of an Oregon sensor to 16-bit by using the channel code of the sensor as the upper byte and the normal sensor ID as the lower byte. The default is NO, to use only the normal sensor ID.

ENGINE_POLL directive

This directive sets the delay in the polling loop of the Heyu Engine and Heyu Monitor daemons. Too low a value results in somewhat excessive CPU usage when idle. The value is expressed in microseconds, with the default being 10000 (= 10 milliseconds). The valid range is 100 to 100000 microseconds.
Example:
ENGINE_POLL 10000

RESTRICT_DIMS directive

The values for dim and bright levels for both Direct and Macro commands is specified in the X10 protocol as ranging from 1-22, however the field in the binary command can accept a number between 0-31. It is unknown at this time whether additional dims/brights beyond 210 are actually sent by the CM11A, however the time for execution of the command is observed to increase approximately proportionately. Setting this directive to NO allows values between 0 and 31 to be programmed without generating an error message, while taking the default of YES restricts the range to 1-22. In any event, a CM11A receiving Dim or Bright signals over the AC power line never reports an expanded value greater than 210, which is actually reached when the programmed level is 19 or greater. (Entering a value of 0 for dim/bright level for my CM11As appears to send the same number of dims/brights as level 2, but who knows what yours will do.)

POWERFAIL_UPDATE directive

When AC power resumes after an interruption, the CM11A will poll the computer with a constant stream of requests at one-second intervals for a time update and will accept no other command until the request is satisfied. The Heyu relay by default automatically responds and updates the CM11A clock. The choices for this directive are either YES or NO, with the default being YES. Choosing NO for this directive instructs the relay to respond with a truncated update which satisfies the CM11A but doesn´t actually reset the clock.

Note: If Heyu is already running, use ´heyu restart´ for changes in this directive to become effective in the heyu_relay daemon.

This directive allows the user to determine the accuracy of the CM11A clock when operating solely on battery power. E.g., run ´heyu setclock´ to set the CM11A clock to the system clock, unplug the CM11A from the wall socket for several hours or a day, plug it back in and run ´heyu readclock´ to compare the CM11A and system clocks.

AUTO_WAIT directive

This directive instructs Heyu to ping the CM11A before sending each X10 command. It has been reported that the USB port on some non-PC systems running Linux (toasters, maybe?) seem to need this nudge to wake them up. This directive takes a parameter 0 through 300 representing the maximum number of pings (at one second intervals) to try before a ping response is received from the CM11A. The default is 0, meaning the CM11A is not pinged.

ENVIRONMENT

X10CONFIG - Points to a fully qualified file name of your configuration file, if located elsewhere than in one of the standard places.
HEYUSUB - Optionally specifies an additional subdirectory level under the standard places where the configuration file will be found, i.e.,
$HOME/.heyu/$HEYUSUB/
/etc/heyu/$HEYUSUB/

AUTHORS

Re-written to use the CM11A interface by Daniel B. Suthers (dbs@tanj.com).

Originally written (Known as X10) by Larry Campbell (maynard!campbell). System V port, ID file, improved display formats, and other cleanup by John Chmielewski (rogue!jlc). Module aliasing, sunset and sunrise additions by Paul Fox (pgf@foxharp.boston.ma.us). Additional functionality for uploaded timers and macros and for scripts by Charles W. Sullivan (cwsulliv01@heyu.org)

SEE ALSO

http://www.heyu.org
date(1), heyu(1), x10sched(5), x10scripts(5), x10cm17a(5), x10aux(5)